

On Quantum Noncompression

SAMUEL EPSTEIN

Abstract

This article presents a quantum transmission problem, in which Alice is trying to send a number of qbits to Bob. Alice has access to two channels, one that sends classical bits and another that sends quantum bits. We show that under certain error terms, Alice can optimize transmission, up to logarithmic precision, by sending only a classical description of the qbits.

1 Introduction

In a canonical algorithmic information theory example, Alice wants to send a single text message x to Bob. Alice sends a program p to B such that $x = U(p)$, where U is a fixed universal Turing machine. The cost of the transmission is the length of p . Alice can minimize cost by sending $\mathbf{K}(x)$ bits to Bob, where \mathbf{K} is the Kolmogorov complexity function. We now look at the quantum case. Suppose that Alice wants to send a (pure) N qubit quantum state $|\psi\rangle$ to Bob, represented as an normalized vector in a complex vector space, $\mathbb{C}^{2^{\otimes N}}$. Thus given the basis sets $|\beta_n\rangle$, a pure state can be represented as $|\psi\rangle = \sum_n c_n |\beta_n\rangle$, where $\sum_n c_n^* c_n = 1$. Alice has access to two channels, a quantum channel and a classical channel. Alice wants to send $|\psi\rangle$ to Bob. Actor can choose to send M (possibly entangled) qubits $|\theta\rangle$ on the quantum channel and L regular bits p on the classical channel, describing a quantum circuit, representing an encoding of unitary operation V , where $U(p) = \langle V \rangle$. Bob, upon receiving $|\theta\rangle$ and p , constructs the unitary operation V , and then applies it to $|\theta\rangle$ (with padded zeros) to produce $|\psi'\rangle = V|\theta 0^{N-M}\rangle$. Bob is not required to produce $|\psi\rangle$ exactly. Instead the fidelity of the attempt is measured by $F = -\log |\langle \psi | \psi' \rangle|^2$. Alice's goal is to minimize $\text{Cost}(|\theta\rangle, p) = L + M + F$. The main result and proofs of this paper are in support of the following statement:

Statement. *For non-exotic quantum states $|\psi\rangle$, Alice can minimize the above cost function up to logarithmic precision by leaving the quantum channel empty, i.e $|\theta\rangle = |0^N\rangle$.*

A state is exotic if it has high mutual information with the halting sequence. The above claim assumes that Alice is trying to communicate a *known* quantum state to Bob. This state could be generated randomly, i.e. Alice has access to a known model, (i.e. a mixed quantum state ρ), and access to an arbitrarily long sequence of random bits. Alice then randomly generates a pure state $|\psi\rangle$ from ρ , encoded in a natural form as an infinite string, $\text{Enc}(|\psi\rangle)$. At this point Alice must communicate $|\psi\rangle$ to Bob, and she has access to a classical channel and a quantum channel. Alice can minimize the above Cost function, up to logarithmic precision, by ignoring the quantum channel and sending a description on the classical channel. There is an extra cost of $\mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H})$ bits (classical or quantum), which is the mutual information of $\text{Enc}(|\psi\rangle)$ and the halting sequence, \mathcal{H} . For non-exotic states, this term is negligible.

Due to information conservation laws, there is no (randomized) method to produce sequences (i.e. $\text{Enc}(|\psi\rangle)$) with arbitrarily high mutual information with the halting sequence. Thus there is no way Alice can generate a quantum state that isn't optimally approximated with a classical description of a state.

2 Related Work

The study of Kolmogorov complexity originated from the work of [Kolmogorov 1965]. The canonical self-delimiting form of Kolmogorov complexity was introduced in [Zvonkin and Levin 1970] and [Chaitin 1975]. The universal probability \mathbf{m} was introduced in [Solomonoff 1964]. More information about the history of the concepts used in this paper can be found in the textbook [Li and Vitányi 2008]. A history of the origin of the mutual information of a string with the halting sequence can be found in [Vereshchagin and Vitányi 2004b].

At a Tallinn conference in 1973, Kolmogorov formulated notion of a two part code and introduced the structure function (see [Vereshchagin and Vitányi 2004b] for more details). Related aspects involving stochastic objects were studied in [Shen 1983, 1999, V'Yugin 1987]. The work of Kolmogorov and the modelling of individual strings using a two-part code was expanded upon in [Vereshchagin and Vitányi 2004b, Gács, Tromp, and Vitányi 2001]. These works introduced the notion of using the prefix of an encoding of Chaitin's halting probability to define a universal algorithmic sufficient statistic of strings.

The generalization and synthesis of this work and the development of algorithmic statistics can be seen in the works of [Vereshchagin and Vitányi 2004a, Vereshchagin and Vitányi 2010, Vereshchagin 2013]. This led to the central theorem in [Epstein and Levin 2011-2014, Epstein 2013, Levin 2014], which showed that the algorithmic probability of non-exotic finite sets of strings are dominated by their simplest element. The first game theoretic proof this theorem can be found in [Shen 2012]. This theorem is a followup to the ϵ term of theorem 6 in [Vereshchagin and Vitányi 2004a] (and its variant in [Epstein and Betke 2011]), which states that if a string is contained by a large number of sets of bounded complexity, then it is contained by a simple set.

Quantum algorithmic probability was studied in [Gács 2001]. A type of quantum complexity dependent on descriptive complexity was introduced in [Vitányi 1999]. Another variant of quantum complexity, as seen in [Berthiaume, van Dam, and Laplante 2000], uses a universal quantum Turing machine.

3 Conventions

We use \mathbb{R} , \mathbb{Q} , \mathbb{N} , \mathbb{W} , \mathbb{C} , $\{0, 1\}^*$, and $\{0, 1\}^\infty$ to represent reals, rationals, natural numbers, whole numbers, complex numbers, finite strings, and infinite strings. Let $X_{\geq 0}$ and $X_{> 0}$ be the sets of non-negative and of positive elements of X . The length of a string $x \in \{0, 1\}^n$ is denoted by $\|x\| = n$. The removal of the last bit of a string is denoted by $(p0^-) = (p1^-) = p$, for $p \in \{0, 1\}^*$. For the empty string \emptyset , (\emptyset^-) is undefined. We use $\{0, 1\}^{*\infty}$ to denote $\{0, 1\}^* \cup \{0, 1\}^\infty$, the set of finite and infinite strings. The i th bit of a string $x \in \{0, 1\}^{*\infty}$ is denoted by x_i . The indicator function of a mathematical statement A is denoted by $[A]$, where if A is true then $[A] = 1$, otherwise $[A] = 0$. The size of a finite set S is denoted to be $|S|$. As is typical of the field of algorithmic information

theory, the theorems in this paper are relative to a fixed universal machine, and therefore their statements are only relative up to additive and logarithmic precision.

The terms $<^+f$, $>^+f$, $=^+f$ represent $<f+O(1)$, $>f-O(1)$, and $=f\pm O(1)$, respectively. For nonnegative real function f , the terms $<^{\log}f$, $>^{\log}f$, $=^{\log}f$ represent the terms $<f+O(\log(f+1))$, $>f-O(\log(f+1))$, and $=f\pm O(\log(f+1))$, respectively. A discrete measure is a nonnegative function $Q : \mathbb{W} \rightarrow \mathbb{R}_{\geq 0}$ over whole numbers. The support of a measure Q is the set of all elements $a \in \mathbb{W}$ that have positive measure, with $\text{Supp}(Q) = \{a : Q(a) > 0\}$. The mean of a function $f : \mathbb{W} \rightarrow \mathbb{R}$ by a measure Q is denoted by $\mathbf{E}_Q[f] = \sum_{a \in \mathbb{W}} f(a)Q(a)$. We say Q is a semimeasure iff $\mathbf{E}_Q[1] \leq 1$. Furthermore, we say that Q is probability measure iff $\mathbf{E}_Q[1] = 1$. The image of a measure Q with respect to a (partial) function $f : \mathbb{W} \rightarrow \mathbb{W}$ is defined to be $(fQ)(x) = \sum \{Q(y) : f(y) = x, y \in \mathbb{W}\}$. If Q is a semimeasure then fQ is also semimeasure (and analogously for probability measures).

3.1 Self Delimiting Codes

The prefix operator over two strings is denoted by \sqsubseteq , where for finite string $x \in \{0, 1\}^*$ and arbitrary string $y \in \{0, 1\}^{\infty}$, we say $x \sqsubseteq y$ iff there exists some $z \in \{0, 1\}^{\infty}$ such that $xz = y$. Furthermore, $x \sqsubseteq x$ for $x \in \{0, 1\}^{\infty}$. When it is clear from the context, we will use whole numbers and other finite objects interchangeably with their binary representations. For example, each whole number $n \in \mathbb{W}$ can be associated with the $(n+1)$ th string of a length increasing lexicographical ordering $\{\xi_n\}_{n=1}^{\infty}$, $\xi_n \in \{0, 1\}^*$, with

$$(0, 0), (1, 1), (2, 00), (3, 01), (4, 10), (5, 11), (6, 000) \dots$$

Thus $\xi_6 = 000$. A prefix free set of codes $S \subset \{0, 1\}^*$ is a set of strings such that there does not exist two distinct strings x, y in S where one string is a prefix of the other, $x \sqsubseteq y$. By the Kraft inequality, for such a prefix free set S of strings,

$$\sum_{x \in S} 2^{-\|x\|} \leq 1.$$

We say such S is a *self-delimiting* code because there exists a method to determine where each code word $x \in S$ ends without reading past its last symbol. One such code word is $\langle x \rangle' = 1^{\|x\|}0x$, where the decoding algorithm would first count the number of 1's before the first 0 to determine the length of x and then output the $\|x\|$ remaining bits in the input, (corresponding to x). Thus $\|\langle x \rangle'\| = 2\|x\| + 1$. For a finite string x , we use $\langle x \rangle \in \{0, 1\}^*$ to denote a more efficient code, with $\langle x \rangle = \langle \xi_{\|x\|} \rangle' 0x$. For example, $\langle 11111 \rangle = 11011011111$. Thus $\|\langle x \rangle\| \leq \|x\| + 2\lceil \log \|x\| \rceil + 2$.

The encoding of a rational number $r \in \mathbb{Q}$ is defined to be $\langle r \rangle = \langle p, q \rangle$ for reduced $p/q = r$. The encoding of a finite set $\{x_n\}_{n=1}^m$ of strings is defined to be $\langle \{x_1, \dots, x_m\} \rangle = \langle m \rangle \langle x_1 \rangle \dots \langle x_m \rangle$, and it is also denoted as $\langle x_1, \dots, x_m \rangle$. $\langle x, \alpha \rangle = \langle x \rangle \alpha$ for $x \in \{0, 1\}^*$ and $\alpha \in \{0, 1\}^{\infty}$. For two infinite strings $\alpha, \beta \in \{0, 1\}^{\infty}$, their encoding is $\langle \alpha, \beta \rangle = \alpha_1 \beta_1 \alpha_2 \beta_2 \alpha_3 \beta_3 \dots$. The terms $\text{Dom}(F)$ and $\text{Range}(F)$ denote the domain and range of a function F . If a function F from whole numbers to whole numbers has a finite domain, then its self delimiting code is denoted by $\langle F \rangle = \langle \{(a, F(a)) : a \in \text{Dom}(F)\} \rangle$. We call such functions, *primitive maps*. A measure $Q : \mathbb{W} \rightarrow \mathbb{Q}_{\geq 0}$ with a finite support and a range of nonnegative rational numbers is called a *primitive measure*, and its self delimiting code is denoted by $\langle Q \rangle = \langle \{(a, Q(a)) : a \in \text{Supp}(Q)\} \rangle$. Both primitive maps and measures admit a finite explicit description.

3.2 Algorithms.

Our paper uses self-delimiting machines M which have four tapes: a main input tape, an auxiliary input tape, a work tape, and an output tape. The alphabet for all tape is $\{0, 1, \$\}$. We say that M computes the partial function $T : \{0, 1\}^* \otimes \{0, 1\}^{*\infty} \rightarrow \{0, 1\}^*$, if whenever $T_\alpha(x)$ is defined, M outputs $y = T_\alpha(x)$ when given $x \in \{0, 1\}^*$ and $\alpha \in \{0, 1\}^{*\infty}$ as input. More specifically,

1. M starts with all its heads in the leftmost square. The main input tape starts with $x\$^\infty$. The auxiliary tape is set to α if it is an infinite string, otherwise it starts with $\alpha\$^\infty$. The work and output tape start with $\$^\infty$.
2. During its operation, M reads only $\|x\|$ bits from the main input tape.
3. The output tape is $y\$^\infty$ when M halts.

When inputs x and α are not defined for T , we say $T_\alpha(x) = \perp$ and the machine M does not perform the steps enumerated above. We say that a partial function T is self delimiting, or prefix free, if there is a self delimiting machine that computes it. The domain of such T is prefix free, where for all $x, y \in \{0, 1\}^*$, $\alpha \in \{0, 1\}^{*\infty}$, with $y \neq \emptyset$, it must be that $T_\alpha(x) = \perp$ or $T_\alpha(xy) = \perp$. For convenience, we use symbols to interchangeably to denote both the machines and also the partial functions (between finite strings) they compute.

A continuous map from infinite sequences to finite or infinite sequences, $\gamma : \{0, 1\}^\infty \rightarrow \{0, 1\}^{*\infty}$ is computable iff there exists a self-delimiting machine M that on input $\alpha \in \{0, 1\}^\infty$, either (1): outputs, without halting, $\beta = \gamma(\alpha)$ on the output tape if $\beta \in \{0, 1\}^\infty$, or (2): outputs $\beta \in \{0, 1\}^*$ on the output tape and halts.

We use a fixed universal prefix-free machine U , where for each prefix-free machine T , there exists $t \in \{0, 1\}^*$ where $U_\alpha(tx) = T_\alpha(x)$ for all $x \in \{0, 1\}^*$ and $\alpha \in \{0, 1\}^{*\infty}$. One example is for such t to be equal to $\langle i \rangle$, where i is the first index of T in an enumeration of self-delimiting machines. A set of whole numbers is (recursively) enumerable if it is the range of a partial recursive function. We say that a real valued function $f : \mathbb{W} \rightarrow \mathbb{R}$ over whole numbers is upper semi-computable if the set $\{(a, q) : f(a) < q \in \mathbb{Q}\}$ is enumerable. We say f is lower semi-computable if $-f$ is upper semi-computable. We say that a program $p \in \{0, 1\}^*$ computes a map $\gamma : \{0, 1\}^\infty \rightarrow \{0, 1\}^{*\infty}$ if $U_\alpha(p) = \gamma(\alpha)$ for all $\alpha \in \{0, 1\}^\infty$.

3.3 Complexity.

The Kolmogorov complexity of string $x \in \{0, 1\}^*$ relative to string $\alpha \in \{0, 1\}^{*\infty}$ is defined to be $\mathbf{K}(x|\alpha) = \min\{\|p\| : U_\alpha(p) = x\}$. We will also use the term algorithmic entropy interchangeably with Kolmogorov complexity. The Solomonoff prior is defined to be the probability that U outputs x when given random bits as input, with $\mathbf{m}(x|\alpha) = \sum_{p: U_\alpha(p)=x} 2^{-\|p\|}$.

The function \mathbf{m} is a universal lower semi-computable semi-measure, in that for any lower semi-computable semi-measure p , there exists $c_p \in \mathbb{N}$ such that for all $x \in \{0, 1\}^*$, one has that $p(x) < c_p \mathbf{m}(x)$. By the coding lemma, for all $x \in \{0, 1\}^*$, one gets that $\mathbf{K}(x) =^+ -\log \mathbf{m}(x)$. The halting sequence $\mathcal{H} \in \{0, 1\}^\infty$ is the characteristic sequence of the domain of U . Chaitin's omega $\Omega \in \mathbb{R}$ is the probability that the universal machine U will halt when given random bits as input, with $\Omega = \sum_{a \in \mathbb{W}} \mathbf{m}(a)$. The mutual information of two strings x and y is $\mathbf{I}(x : y) = \mathbf{K}(x) + \mathbf{K}(y) - \mathbf{K}(x, y)$. The information that the halting sequence \mathcal{H} has about x (or any finite object it encodes), conditional to $\alpha \in \{0, 1\}^{*\infty}$ is denoted by $\mathbf{I}(x; \mathcal{H}|\alpha) = \mathbf{K}(x|\alpha) - \mathbf{K}(x|\alpha, \mathcal{H})$. This is the difference of the

algorithmic entropy of x and the algorithmic entropy x relativized to \mathcal{H} (with both complexities also relativized to α). We define information with respect to two infinite sequences $\alpha, \beta \in \{0, 1\}^\infty$ as $\mathbf{I}(\alpha : \beta) = \log \sum_{x, y \in \{0, 1\}^*} \mathbf{m}(x|\alpha) \mathbf{m}(y|\beta) 2^{\mathbf{I}(x:y)}$. This function follows algorithmic conservation laws.

Theorem 1 (Information Non-growth).

1. For program $p \in \{0, 1\}^*$ that computes the map $\gamma : \{0, 1\}^\infty \rightarrow \{0, 1\}^\infty$, for all $\alpha, \beta \in \{0, 1\}^\infty$,

$$\mathbf{I}(\gamma(\alpha) : \beta) <^+ \mathbf{I}(\alpha : \beta) + \mathbf{K}(p).$$

2. For $x \in \{0, 1\}^*$ and $\alpha \in \{0, 1\}^\infty$,

$$\mathbf{I}(x; \mathcal{H}) <^+ \mathbf{I}(\alpha : \mathcal{H}) + \mathbf{K}(x|\alpha).$$

Proof. For (1), by the definition of computable maps and their programs, we have that for all $x \in \{0, 1\}^*$, $\mathbf{K}(x|\alpha) <^+ \mathbf{K}(x|\gamma(\alpha)) + \mathbf{K}(p)$. This is because a program $q \in \{0, 1\}^*$ to compute x from $\gamma(a)$ can be combined with p to create a program $r = r_0 p^* q$ that will compute x from α . p^* is the shortest program to output p , and r_0 is helper code that runs p^* and then feeds the auxilliary input through p and then q . So we have that

$$\begin{aligned} \mathbf{I}(\gamma(\alpha) : \mathcal{H}) &= \log \sum_{x, y} \mathbf{m}(x|\gamma(\alpha)) \mathbf{m}(y|\beta) 2^{\mathbf{I}(x:y)} \\ &<^+ \log \sum_{x, y} \mathbf{m}(x|\alpha) \mathbf{m}(y|\beta) 2^{\mathbf{I}(x:y)} / \mathbf{m}(p) \\ &=^+ \mathbf{K}(p) + \log \sum_{x, y} \mathbf{m}(x|\alpha) \mathbf{m}(y|\beta) 2^{\mathbf{I}(x:y)} \\ &<^+ \mathbf{K}(p) + \mathbf{I}(\alpha : \beta). \end{aligned}$$

For (2) we have the inequalities,

$$\begin{aligned} \mathbf{I}(\alpha : \mathcal{H}) &> \log \sum_y \mathbf{m}(y|\alpha) \mathbf{m}(y|\mathcal{H}) 2^{\mathbf{I}(y:y)} \\ &> \log \mathbf{m}(x|\alpha) \mathbf{m}(x|\mathcal{H}) 2^{\mathbf{I}(x:x)} \\ &=^+ \log \mathbf{m}(x|\alpha) \mathbf{m}(x|\mathcal{H}) 2^{\mathbf{K}(x)} \\ &=^+ -\mathbf{K}(x|\alpha) + \mathbf{K}(x) - \mathbf{K}(x|\mathcal{H}). \\ \mathbf{I}(\alpha : \mathcal{H}) + \mathbf{K}(x|\alpha) &>^+ \mathbf{I}(x; \mathcal{H}). \end{aligned}$$

□

We will leverage the following theorem in section 4. The proof of this theorem can be found in appendix A.

Theorem 2.

For primitive map f , $\min_{a \in \text{Dom}(f)} f(a) + \mathbf{K}(a) <^{\log} -\log \sum_{a \in \text{Dom}(f)} \mathbf{m}(a) 2^{-f(a)} + \mathbf{I}(\langle f \rangle; \mathcal{H})$.

4 Quantum

For a complex vector $c = a + bi$, its conjugate is represented by $c^* = a - bi$, and we say that $|c|^2 = c^*c = a^2 + b^2$. The complex vector c is primitive if its real and imaginary components are rational, and its encoding is denoted by $\langle c \rangle = \langle a, b \rangle$. We deal with a finite number N qubits, and we use the Hilbert space $\mathbb{H}_N = (\mathbb{C}^2)^{\otimes N}$. This is a complex vector space of dimension 2^N , where the scalar field is the set of complex numbers, \mathbb{C} . Any vector is a linear combination of the orthonormal basis set $\{|\beta_n\rangle\}$. Pure states $|\psi\rangle$ are represented as unit vectors in this space, with $|\psi\rangle = \sum_n c_n |\beta_n\rangle$ and $\sum_n |c_n|^2 = 1$. The set of pure states is represented by \mathcal{P}_N . For a pure state $|\theta\rangle \in \mathcal{P}_M$ of $M \leq N$ qubits, $|\theta 0 \dots\rangle$ is used to denote $|\theta\rangle |0^{N-M}\rangle$. The bra $\langle\psi|$, is a unit vector of the dual space of \mathbb{H}_N .

Mixed states are represented as a probability distribution $\{p_n\}$ over a set of pure states, $\{|\psi_n\rangle\}$, where $\sum_n p_n = 1$. A Hermitian matrix A , is a square $2^N \times 2^N$ matrix that is equal to its own conjugate transpose, where $a_{nm} = a_{mn}^*$. Unless otherwise stated, the dimensions of the matrices in this section are $2^N \times 2^N$. The conjugate transpose of a matrix is denoted by A^* . The eigenvalues of a Hermitian matrix are always real. A Hermitian matrix A is called nonnegative, $A \succeq 0$, if all its eigenvalues are nonnegative. For Hermitian matrices A and B , we say $A \succeq B$ iff $(A - B) \succeq 0$. If $A \succeq B$ then $\langle\psi|A|\psi\rangle \geq \langle\psi|B|\psi\rangle$ for all $|\psi\rangle \in \mathcal{P}_N$.

The trace of a matrix A is represented as $\text{Tr } A = \sum_n \langle\beta_n|A|\beta_n\rangle$. Density matrices ρ are non-negative Hermitian matrices of trace = 1. Pure states $|\psi\rangle$ have a dual representation as density matrices $\rho = |\psi\rangle\langle\psi|$. Mixed states $\{p_n\}$ over a set of pure states $\{|\psi_n\rangle\}$ have a dual representation as density matrices, with $\sum_n p_n |\psi_n\rangle\langle\psi_n|$.

A semi density matrix is a nonnegative Hermitian matrix of trace ≤ 1 . Pure states are represented using $|\psi\rangle$, and (semi)-density matrices are represented using ρ . A pure state $|\psi\rangle = \sum_n c_n |\beta_n\rangle$ of N qubits is primitive, if all its coefficients c_n are primitive. Such primitive states admit a natural encoding $\langle|\psi\rangle\rangle = \langle\{c_1, \dots, c_{2^N}\}\rangle$. For such primitive states, their algorithmic probability is $\mathbf{m}(|\psi\rangle) = \mathbf{m}(\langle|\psi\rangle\rangle)$. The set of primitive pure N -bit qubit states is \mathcal{R}_N .

The identity matrix is denoted by I . A matrix V is primitive iff all its entries v_{nm} are primitive. Primitive matrices admit a natural encoding, $\langle V \rangle = \langle\{v_{nm} : n, m \in [1, 2^N]\}\rangle$. Transformations between states occur due to unitary matrices V of dimension 2^N . A matrix is unitary iff $V^*V = VV^* = I$.

We now describe an (infinite) encoding scheme for an arbitrary quantum pure state $|\psi\rangle$. This scheme is defined as an injection between \mathcal{P}_N and $\{0, 1\}^\infty$. We define $\text{Enc} : \mathcal{P}_N \rightarrow \{0, 1\}^\infty$ to be an injection, where $\text{Enc}(|\psi\rangle)$ is an ordered list of the encoded tuples $\langle|\theta\rangle, q, [|\langle\psi|\theta\rangle|^2 \geq q]\rangle$, over all primitive states $|\theta\rangle \in \mathcal{R}_N$ and rational distances $q \in \mathbb{Q}_{>0}$.

Proposition 4.1. *There is a computable map between infinite sequences such that for any primitive unitary transform V and pure state $|\psi\rangle \in \mathcal{P}_N$, one has that $\langle V \rangle \text{Enc}(|\psi\rangle) \mapsto \text{Enc}(V|\psi\rangle)$.*

Proof. One can determine $b \in \{0, 1\}$ for each encoded tuple $\langle|\theta\rangle, q, b\rangle$ in $\text{Enc}(V|\psi\rangle)$ by examining an encoded tuple in the original sequence $\text{Enc}(|\psi\rangle)$. In particular

$$b = [|\langle\psi|V^*|\theta\rangle|^2 \geq q] = [|\langle\psi|(V^*|\theta\rangle)|^2 \geq q].$$

Thus b is equal to the bit b' of the tuple $(V^*|\theta\rangle, q, b')$. This tuple is encoded in $\text{Enc}(|\psi\rangle)$, since $V^*|\theta\rangle$ is primitive and q is rational. \square

We say that mutual information, between pure state $|\psi\rangle$ and the halting sequence is defined to be equal to mutual information between the infinite sequence $\text{Enc}(|\psi\rangle)$ and \mathcal{H} , with $\mathbf{I}(|\psi\rangle : \mathcal{H}) = \mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H})$. This function obeys information non-growth laws.

Lemma 4.2 (Information Non-growth).

For primitive unitary transform V and pure state $|\psi\rangle$, $\mathbf{I}(V|\psi) : \mathcal{H}) <^+ \mathbf{I}(|\psi) : \mathcal{H}) + \mathbf{K}(V)$.

Proof. For all $x \in \{0,1\}^*$, one has that $\mathbf{K}(x|\text{Enc}(|\psi\rangle)) <^+ \mathbf{K}(x|\text{Enc}(V|\psi)) + \mathbf{K}(V)$. This is due to the following reasoning. Let p be a shortest program that prints x when $\text{Enc}(V|\psi)$ is on the auxilliary tape. Then there exists a program $p' = p_0 v^* p$ that prints x when $\text{Enc}(|\psi\rangle)$ is on the auxilliary tape. The string v^* is a shortest program that outputs $\langle V \rangle$. The string p_0 is helper code that applies V to the auxilliary tape (using the reasoning of proposition 4.1), and then sends the output to the program p . The output of p is set as the output of the program p' . Thus $\mathbf{m}(x|\text{Enc}(V|\psi)) \leq O(1)\mathbf{m}(x|\text{Enc}(|\psi\rangle))/\mathbf{m}(V)$. So

$$\begin{aligned} \mathbf{I}(V|\psi) : \mathcal{H}) &= \log \sum_{x,y} \mathbf{m}(x|\text{Enc}(V|\psi)) \mathbf{m}(y|\mathcal{H}) 2^{\mathbf{I}(x:y)} \\ &<^+ \mathbf{K}(V) + \log \sum_{x,y} \mathbf{m}(x|\text{Enc}(|\psi\rangle)) \mathbf{m}(y|\mathcal{H}) 2^{\mathbf{I}(x:y)} \\ &<^+ \mathbf{K}(V) + \mathbf{I}(|\psi) : \mathcal{H}). \end{aligned}$$

□

We use a modified version of the entropy term in [GacsSomething]. The main difference is the absense of the hidden parameter N . The Gács entropy of a pure state $|\psi\rangle$ of N qubits is $\mathbf{Hg}(|\psi\rangle) = -\log \sum_{|\theta\rangle \in \mathcal{R}_N} \mathbf{m}(|\theta\rangle) \langle \psi|\theta\rangle \langle \theta|\psi\rangle$. We use $\boldsymbol{\mu} = \sum_{|\theta\rangle \in \mathcal{R}_N} \mathbf{m}(|\theta\rangle) |\theta\rangle \langle \theta|$. The Vitanyi entropy of a pure state $|\psi\rangle$ is the classical encoding of a pure state plus the fidelity error term, with $\mathbf{Hv}(|\psi\rangle) = \min\{\mathbf{K}(|\theta\rangle) - \log |\langle \psi|\theta\rangle|^2 : |\theta\rangle \in \mathcal{R}_N\}$. Another method to define the information content of a pure quantum state is via quantum circuits. A quantum circuit (V, M) consists of two parts, (1) a primitive unitary transform from \mathcal{P}_N to \mathcal{P}_N , and (2) the number of qbits $M \leq N$ that are inputted into the circuit, with the rest $N - M$ bits being hardcoded to 0. For an input $|\theta\rangle \in \mathcal{P}_M$, the quantum circuit produces the output $|\theta\rangle \mapsto V|\theta 0.. \rangle$. The set of all quantum circuits over N qbits is denoted by \mathcal{C}_N .

The circuit entropy \mathbf{Hc} of a pure state $|\psi\rangle$ can be thought of as an information transmission problem between Alice and Bob. Alice first send a description of the quantum circuit to Bob, and then Alice sends a number of entangled qbits to Bob. The cost is the sum of the number of bits to describe the circuit, plus the number of qbits that is input into the circuit, plus an error term. Thus $\mathbf{Hc}(|\psi\rangle) = \min\{\mathbf{K}(V, M) + M - \log |\langle \psi|V|\theta 0.. \rangle|^2 : (V, M) \in \mathcal{C}_N, |\theta\rangle \in \mathcal{P}_M\}$. Thus to recap, we have the following definitions, with:

$$\begin{aligned} \text{Gacs Entropy: } \quad & \mathbf{Hg}(|\psi\rangle) = -\log \sum \{\mathbf{m}(|\theta\rangle) \langle \psi|\theta\rangle \langle \theta|\psi\rangle : |\theta\rangle \in \mathcal{R}_N\}, \\ \text{Vitanyi Entropy: } \quad & \mathbf{Hv}(|\psi\rangle) = \min\{\mathbf{K}(|\theta\rangle) - \log |\langle \psi|\theta\rangle|^2 : |\theta\rangle \in \mathcal{R}_N\}, \\ \text{Circuit Entropy: } \quad & \mathbf{Hc}(|\psi\rangle) = \min\{\mathbf{K}(V, M) + M - \log |\langle \psi|V|\theta 0.. \rangle|^2 : (V, M) \in \mathcal{C}_N, |\theta\rangle \in \mathcal{P}_M\}. \end{aligned}$$

4.1 Equivalence of Entropies

We show that for non-exotic pure states, i.e. $|\psi\rangle$ with low $\mathbf{I}(|\psi\rangle : \mathcal{H})$, the entropy terms listed above are more or less equivalent.

Lemma 4.3. *For each primitive semi-density matrix A , one has that $\mathbf{m}(A)A \preceq O(1)\mu$.*

Proof. The matrix A admits an eigenvalue/eigenvector decomposition $A = \sum_{n=1}^{2^N} \lambda_n |\psi_n\rangle\langle\psi_n|$. Since A is a semi-density matrix, one has that $\sum_{n=1}^{2^N} \lambda_n \leq 1$. Also since A is primitive, each $|\psi_n\rangle$ is primitive. Thus given A , one can identify each $|\psi_n\rangle$ with a code of length $<^+ -\log \lambda_n$. Thus $\mathbf{K}(|\psi\rangle) <^+ \mathbf{K}(A) - \log \lambda_n$, which implies that $\mathbf{m}(A)\lambda_n \leq O(1)\mathbf{m}(|\psi_n\rangle)$. Thus $\mathbf{m}(A)A = \mathbf{m}(A) \sum_n \lambda_n |\psi_n\rangle\langle\psi_n| \leq O(1) \sum_n \mathbf{m}(|\psi_n\rangle) |\psi_n\rangle\langle\psi_n| \leq O(1)\mu$. \square

Lemma 4.4. *For all $|\psi\rangle \in \mathcal{P}_N$, $\mathbf{Hc}(|\psi\rangle) <^{\log} \mathbf{Hv}(|\psi\rangle)$*

Proof. Let $|\theta\rangle \in \mathcal{R}_N$ minimize $\mathbf{Hv}(|\psi\rangle)$, and let $c = -\log [|\langle\psi|\theta\rangle|^2]$. Given $(|\theta\rangle, c)$, one can construct a circuit $(V, 0)$ such that $V|0^N\rangle = |\theta\rangle$ (see the Gram Schmidt procedure). Thus $\mathbf{K}(|\theta\rangle) <^+ \mathbf{K}(V)$. Thus $\mathbf{Hc}(|\psi\rangle) <^+ \mathbf{K}(V, 0) + 0 - \log |\langle\psi|V|0^N\rangle|^2 <^+ \mathbf{K}(|\theta\rangle, c) + c <^{\log} \mathbf{K}(|\theta\rangle) + c =^{\log} \mathbf{Hv}(|\psi\rangle)$. \square

Lemma 4.5. *For all $|\psi\rangle \in \mathcal{P}_N$, $\mathbf{Hg}(|\psi\rangle) <^+ \mathbf{Hv}(|\psi\rangle)$.*

Proof. The first inequality follows because $\mathbf{Hg}(|\psi\rangle) = -\log \sum_{|\theta\rangle \in \mathcal{R}_N} \mathbf{m}(|\theta\rangle) \langle\psi|\theta\rangle \langle\theta|\psi\rangle$ is a negative-log sum function, whereas $\mathbf{Hv}(|\psi\rangle)$ can be written, up to additive constants, as a negative-log max function, with $\mathbf{Hv}(|\psi\rangle) =^+ -\log \max_{|\theta\rangle \in \mathcal{R}_N} \mathbf{m}(|\theta\rangle) \langle\psi|\theta\rangle \langle\theta|\psi\rangle$. \square

Theorem 3. *For all $|\psi\rangle \in \mathcal{P}_N$, $\mathbf{Hg}(|\psi\rangle) <^+ \mathbf{Hc}(|\psi\rangle)$.*

Proof. For a circuit (V, M) , let $\gamma(V, M) = 2^{-M} V I_M V^*$ denote the primitive density matrix applying V to the projector I_M of all M length qubit inputs. So $\mathbf{m}(V, M) \gamma(V, M) < O(1) \mathbf{m}(\gamma(V, M)) \gamma(V, M) < O(1)\mu$. The last inequality is due to lemma 4.3. Let the circuit (V, M) and the pure quantum state $|\theta\rangle \in \mathcal{P}_M$ minimize $\mathbf{Hc}(|\psi\rangle)$. Therefore $\mathbf{m}(V, M) 2^{-M} V |\theta 0 \dots\rangle \langle \theta 0 \dots| V^* \preceq \mathbf{m}(V, M) \gamma(V, M) \preceq O(1)\mu$. So

$$\begin{aligned} \mathbf{Hg}(|\psi\rangle) &= -\log \langle\psi|\mu|\psi\rangle \\ &<^+ -\log \mathbf{m}(V, M) 2^{-M} \langle\psi|V|\theta 0 \dots\rangle \langle \theta 0 \dots|V^*|\psi\rangle \\ &=^+ \mathbf{K}(V, M) + M - \log |\langle\psi|V|\theta 0 \dots\rangle|^2 \\ &=^+ \mathbf{Hc}(|\psi\rangle). \end{aligned}$$

\square

Theorem 4. *For all $|\psi\rangle \in \mathcal{P}_N$, $\mathbf{Hv}(|\psi\rangle) <^{\log} \mathbf{Hg}(|\psi\rangle) + \mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H})$.*

Proof. Let $\mathcal{D} \subset \mathcal{R}_N$ be a finite set of primitive vectors, computable from $\text{Enc}(|\psi\rangle)$ and $g = \lceil \mathbf{Hg}(|\psi\rangle) \rceil$ such that $-\log \sum_{|\theta\rangle \in \mathcal{D}} \mathbf{m}(|\theta\rangle) |\langle\psi|\theta\rangle|^2 \leq g + 1$. It is computable because there exists an algorithm that can find \mathcal{D} by the following method. The algorithm enumerates all primitive states $|\theta\rangle \in \mathcal{R}_N$. This algorithm approximates the algorithmic probabilities $\mathbf{m}(|\theta\rangle)$ (from below) with and increasing $\widehat{\mathbf{m}}(|\theta\rangle)$. This algorithm uses $\text{Enc}(|\psi\rangle)$ to approximate $|\langle\theta|\psi\rangle|^2$ from below with $\widehat{|\langle\theta|\psi\rangle|}^2$. This algorithm stops when it finds a finite set $\mathcal{D} \subset \mathcal{R}_N$ such that $-\log \sum_{|\theta\rangle \in \mathcal{D}} \widehat{\mathbf{m}}(|\theta\rangle) \widehat{|\langle\theta|\psi\rangle|}^2 \leq g + 1$.

Thus we have that $\mathbf{K}(\mathcal{D}|g, \text{Enc}(|\psi\rangle)) = O(1)$. Let $f: \mathcal{D} \rightarrow \mathbb{W}$ be a primitive function such that $|\log |\langle \psi | \theta \rangle|^2 - f(|\theta\rangle)| \leq 1$. One such f is computable relative to $\text{Enc}(|\psi\rangle)$, and g . Firstly this is because \mathcal{D} is computable from $\text{Enc}(|\psi\rangle)$ and g . The individual values of f are computable from $\text{Enc}(|\psi\rangle)$, since $|\langle \psi | \theta \rangle|^2$ can be computed to any degree of accuracy. So $\mathbf{K}(f|g, \text{Enc}(|\psi\rangle)) = O(1)$ and $-\log \sum_{|\theta\rangle \in \mathcal{D}} \mathbf{m}(|\theta\rangle) 2^{-f(|\theta\rangle)} \leq g + 2$. One then has that

$$\mathbf{Hv}(|\psi\rangle) <^+ \min_{\theta \in \mathcal{D}} \mathbf{K}(|\theta\rangle) + f(|\theta\rangle) \quad (1)$$

$$<^{\log} -\log \sum_{|\theta\rangle \in \mathcal{D}} \mathbf{m}(|\theta\rangle) 2^{-f(|\theta\rangle)} + \mathbf{I}(\langle f \rangle; \mathcal{H}). \quad (2)$$

$$<^{\log} g + \mathbf{I}(\langle f \rangle; \mathcal{H}) \quad (3)$$

$$<^{\log} g + \mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H}) + \mathbf{K}(\langle f \rangle | \text{Enc}(|\psi\rangle)) \quad (4)$$

$$<^{\log} g + \mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H}) + \mathbf{K}(g)$$

$$<^{\log} \mathbf{Hg}(|\psi\rangle) + \mathbf{I}(\text{Enc}(|\psi\rangle) : \mathcal{H}).$$

Equation 1 is due to the definition of Vitanyi entropy. Inequality 2 is to the main algorithmic statistics theorem of the paper: Theorem 2. Inequality 3 is due to the definition of f and \mathcal{D} . Inequality 4 is due to theorem 1. \square

Theorem 5. $\mathbf{Hv}(|\psi\rangle) <^{\log} \mathbf{Hc}(|\psi\rangle) + \mathbf{I}(|\psi\rangle : \mathcal{H})$.

Proof. This follows from the inequalities

$$\mathbf{Hv}(|\psi\rangle) <^{\log} \mathbf{Hg}(|\psi\rangle) + \mathbf{I}(|\psi\rangle : \mathcal{H}) <^{\log} \mathbf{Hc}(|\psi\rangle) + \mathbf{I}(|\psi\rangle : \mathcal{H}).$$

The first inequality is due to theorem 4 and the second inequality is due to theorem 3. \square

A Algorithmic Statistics.

In this section, we provide the proof for theorem 2. The deficiency of randomness of $x \in \{0, 1\}^*$ with respect to probability measure Q , conditional to $v \in \{0, 1\}^*$, is $\mathbf{d}(x|Q, v) = \lfloor -\log Q(x) \rfloor - \mathbf{K}(x|v)$. Strings x that are typical of a probability measure Q have a small deficiency of randomness, $\mathbf{d}(x|Q) = O(1)$. We say that x is stochastic if it is typical of some simple probability measure. More formally, one says that x is (j, k) stochastic for $j, k \in \mathbb{N}$ if there exists a primitive probability measure Q , with $\langle Q \rangle = U(v)$, $\|v\| = j$, and $\mathbf{d}(x|Q, v) \leq k$. For non-decreasing $\epsilon: \mathbb{N} \rightarrow \mathbb{W}$, we use the stochasticity function $\chi_\epsilon(x) = \min\{j + \epsilon(k) : x \text{ is } (j, k) \text{ stochastic}\}$. For this paper, we use the function $\chi = \chi_\epsilon$ for $\epsilon(d) = 2 \log d$. The conditional stochasticity form¹ is represented by $\chi(x|\alpha)$, for $\alpha \in \{0, 1\}^{*\infty}$. We recall that functions between whole numbers are called *primitive maps* if they have a finite domain. Measures over whole numbers with finite support and with a range containing only nonnegative rational numbers are called *primitive measures*.

Theorem 6. *Let f be a primitive map and m be a primitive probability measure. Let $a \in \mathbb{W}$ vary over $\text{Dom}(f) \cap \text{Supp}(m)$. Then $\min_a f(a) + \mathbf{K}(a|m) <^{\log} -\log \sum_a m(a) 2^{-f(a)} + \chi(f|m)$.*

¹This is formally represented as $\chi(x|\alpha) = \min\{j + \epsilon(k) : \exists v \in \{0, 1\}^j, U_\alpha(v) = \langle Q \rangle, \mathbf{d}(x|Q, \langle v, \alpha \rangle) \leq k\}$.

Proof. Since all terms in the theorem are conditioned on $\langle m \rangle$, we will also condition all complexity terms in the proof on $\langle m \rangle$ and drop its notation. More formally, $U(x)$ is used to denote $U_{\langle m \rangle}(x)$, $\mathbf{K}(x)$ is used to denote $\mathbf{K}(x|m)$, and $\chi(\langle f \rangle)$ is used to denote $\chi(f|m)$.

For any primitive map g , let $g_n = g^{-1}(n) \cap \text{Supp}(m)$ and let $g_{\leq n} = \cup_{i=0}^n g_i$, for $n \in \mathbb{W} \cup \{\infty\}$. Let $s = \lceil -\log \sum_{a \in f_{\leq \infty}} m(a)2^{-f(a)} \rceil$.

$$\sum_{a \in f_{\leq \infty}} m(a)2^{-f(a)} \geq 2^{-s}, \quad (5)$$

$$\sum_{a \in f_{\leq \infty} \setminus f_{\leq s}} m(a)2^{-f(a)} \leq \sum_{a \in f_{\leq \infty} \setminus f_{\leq s}} m(a)2^{-s-1} \leq 2^{-s-1}, \quad (6)$$

$$\sum_{a \in f_{\leq s}} m(a)2^{-f(a)} \geq 2^{-s-1}. \quad (7)$$

Equation (5) follows from the definition of s . Equation (7) follows from equations (5) and (6). We now turn our attention to creating a primitive probability measure Q for which f is typical. Let $v' \in \{0, 1\}^*$ realize the stochasticity of f , with $U(v') = \langle Q' \rangle$, and $\chi(f) = \|v'\| + \epsilon(\max\{\mathbf{d}(f|Q', v'), 1\}) = \|v'\| + 2\log(\max\{\mathbf{d}(f|Q', v'), 1\})$. Note that this implies $\langle f \rangle \in \text{Supp}(Q')$. Let Q be a primitive probability measure² equal to Q' conditioned on the largest set of (encoded) primitive maps g such that $\sum_{a \in g_{\leq s}} m(a)2^{-g(a)} \geq 2^{-s-1}$. Such Q is computable from v' and s . One such Q program $v \in \{0, 1\}^*$, $U(v) = \langle Q \rangle$, is of the form $v = v_0 v_s v'$, where $v_0 \in \{0, 1\}^*$ is helper code of size $O(1)$, and $v_s \in \{0, 1\}^*$ is a shortest U -program for s . So $\|v\| <^+ \|v'\| + \mathbf{K}(s)$. For $d = \max\{\mathbf{d}(f|Q, v), 1\}$ we have that

$$\begin{aligned} \|v\| &<^+ \|v'\| + \mathbf{K}(s), \\ \|v\| + 2\log d &<^+ \|v'\| + \mathbf{K}(s) + 2\log d \\ &<^+ \|v'\| + \mathbf{K}(s) + 2\log(-\log Q(f) - \mathbf{K}(f|v)) \\ &<^+ \|v'\| + \mathbf{K}(s) + 2\log(-\log Q'(f) - \mathbf{K}(f|v)) \\ &<^+ \|v'\| + \mathbf{K}(s) + 2\log(-\log Q'(f) - \mathbf{K}(f|v') + \mathbf{K}(v|v')) \\ &<^{\log} \|v'\| + \mathbf{K}(s) + 2\log(-\log Q'(f) - \mathbf{K}(f|v')), \\ \|v\| + 2\log d &<^{\log} \chi(f) + \mathbf{K}(s). \end{aligned} \quad (8) \quad (9) \quad (10) \quad (11)$$

Equation (8) follows from $Q(g) = [g \in \text{Supp}(Q)]Q'(g)/Q'(\text{Supp}(Q))$, and thus $-\log Q(g) \leq -\log Q'(g)$ for all $g \in \text{Supp}(Q)$. Equation (9) follows from the inequality $\mathbf{K}(f|v') <^+ \mathbf{K}(f|v) + \mathbf{K}(v|v')$. Equation (10) follows from v being computable from v' and v_s , and thus $\mathbf{K}(v|v') <^+ \mathbf{K}(s)$.

Let $c \in \mathbb{N}$ be a constant to be determined later. We use a set of random sets of whole numbers, $\{\lambda_n\}_{n=0}^s$, where $|\lambda_n| = cd2^{s+1-n}$, and for every n , each $a \in \lambda_n$ is chosen independently with probability $m(a)$. For a set of sets $A = \{A_n\}_{n=0}^s$, $A_n \subset \mathbb{W}$, let $\mathbf{1}(g, A) = 1$ if $g_n \cap A_n = \emptyset$ for all

²Thus $Q(\langle g \rangle) = [g \in S]Q'(g)/Q'(S)$, where $S \subset \{0, 1\}^*$, the support of Q , is defined as $S = \{\langle g \rangle : g \in \text{Supp}(Q'), \sum_{a \in g_{\leq s}} m(a)2^{-g(a)} \geq 2^{-s-1}\}$.

$n \in [0, s]$, and $\mathbf{1}(g, A) = 0$, otherwise. Thus

$$\begin{aligned} \mathbf{E}_{g \sim Q} \mathbf{E}_{A \sim \lambda} [\mathbf{1}(g, A)] &= \sum_g Q(g) \prod_{n=0}^s (1 - m(g_n))^{\lambda_n} \\ &\leq \sum_g Q(g) \prod_{n=0}^s \exp\{-|\lambda_n| m(g_n)\} \end{aligned} \quad (12)$$

$$\begin{aligned} &= \sum_g Q(g) \exp\left\{-\sum_{n=0}^s |\lambda_n| m(g_n)\right\} \\ &= \sum_g Q(g) \exp\left\{-\sum_{n=0}^s cd 2^{s+1-n} m(g_n)\right\} \\ &= \sum_g Q(g) \exp\left\{-cd 2^{s+1} \sum_{n=0}^s m(g_n) 2^{-n}\right\}, \\ \mathbf{E}_{g \sim Q} \mathbf{E}_{A \sim \lambda} [\mathbf{1}(g, A)] &\leq \sum_g Q(g) \exp\{-cd\} = \exp\{-cd\}. \end{aligned} \quad (13)$$

Equation (12) follows from the inequality $(1-a) \leq e^{-a}$ over $a \in [0, 1]$. Equation (13) follows from the definition of the support of Q , where $g \in \text{Supp}(Q)$ iff $\sum_{a \in g \leq s} m(a) 2^{-g(a)} \geq 2^{-s-1}$. By the probability argument, there exists a set of sets $A = \{A_n\}_{n=0}^s$ such that $|A_n| = cd 2^{s+1-n}$ and

$$\mathbf{E}_{g \sim Q} [\mathbf{1}(g, A)] \leq \exp\{-cd\}.$$

There exists a brute force search algorithm that on input ³ c, d, v , outputs A . This algorithm computes all possible sets of sets $A' = \{A'_n\}_{n=0}^s$, $|A'_n| = cd 2^{s+1-n}$ and outputs the first A' such that $\mathbf{E}_{g \sim Q} [\mathbf{1}(g, A')] \leq \exp\{-cd\}$. The existence of such an A' is guaranteed by equation (13). We let \hat{Q} be the primitive semimeasure such that $\hat{Q}(g) = \mathbf{1}(g, A) \lfloor e^{cd} \rfloor Q(g)$. \hat{Q} is computable from A, c, d , and v . Thus the above reasoning implies the two inequalities

$$\mathbf{K}(A) <^+ \mathbf{K}(c, d, v), \quad (14)$$

$$\mathbf{K}(\hat{Q}|v) <^+ \mathbf{K}(c, d). \quad (15)$$

There exists $n \in [0, s]$ with $|f_n \cap A_n| > 0$. Otherwise $\hat{Q}(f) = Q(f) \lfloor e^{cd} \rfloor$, causing the following contradiction (for proper choice of c).

$$\begin{aligned} d &\geq \mathbf{d}(f|Q, v) \\ &= \lfloor -\log Q(f) \rfloor - \mathbf{K}(f|v) \\ &> -\log Q(f) - (-\log \hat{Q}(f) + \mathbf{K}(\hat{Q}|v)) - O(1) \end{aligned} \quad (16)$$

$$> -\log Q(f) - (-\log \hat{Q}(f) + \mathbf{K}(c, d)) - O(1) \quad (17)$$

$$> cd \log e - \mathbf{K}(c, d) - O(1) > d. \quad (18)$$

³Note that the strings s and $\langle Q \rangle$ are computable from v .

Equation (16) follows that for any $g \in \text{Supp}(\hat{Q})$, $\mathbf{K}(g) <^+ \mathbf{K}(\hat{Q}) - \log \hat{Q}(g)$. Equation (17) follows from equation (15). Equation (18) follows from the additive equality, $-\log Q(f) + \log \hat{Q}(f) =^+ cd \log e$. The choice of c is dependent solely on U and invariant to conditioned information (as auxiliary strings to U would only decrease $\mathbf{K}(c, d)$). Thus $c \in O(1)$ is removed from consideration for the rest of the proof. For the n defined above, there exists $a \in f_n \cap A_n$ with

$$\begin{aligned} \mathbf{K}(a) &<^+ \log |A_n| + \mathbf{K}(A_n) \\ &<^+ \log |A_n| + \mathbf{K}(A) + \mathbf{K}(A_n|A) \\ &<^+ (\log d + s - n) + \mathbf{K}(d, v) + \mathbf{K}(n) \\ &=^+ \log d + s - f(a) + \mathbf{K}(d, v) + \mathbf{K}(f(a)), \end{aligned} \tag{19}$$

$$\begin{aligned} \mathbf{K}(a) + f(a) &<^+ \log d + s + \mathbf{K}(v) + \mathbf{K}(d) + \mathbf{K}(f(a)), \\ \mathbf{K}(a) + f(a) &<^{\log} s + \|v\| + 2 \log d, \end{aligned} \tag{20}$$

$$\mathbf{K}(a) + f(a) <^{\log} s + \chi(f), \tag{21}$$

$$\min_{a \in f_{\leq \infty}} \mathbf{K}(a) + f(a) <^{\log} -\log \sum_{a \in f_{\leq \infty}} m(a) 2^{-f(a)} + \chi(f). \tag{22}$$

Equation (19) follows from equation (14), and from $c \in O(1)$. Equation (20) follows from $\mathbf{K}(x) <^{\log} \|x\|$ for $x \in \{0, 1\}^* \cup \mathbb{W}$. Equation (21) follows directly from equation (11). Equation (22) follows from the definition of s and its form proves the theorem. \square

A.1 Left-Total Machines

We introduce some more complex notion related to algorithmic statistics for the remaining theorems, in particular the “left-total” universal algorithm. We say $x \in \{0, 1\}^*$ is total with respect to a machine if the machine halts on all sufficiently long extensions of x . More formally, x is total with respect to T_y iff there exists a prefix free set of strings $Z \subset \{0, 1\}^*$ where $\sum_{z \in Z} 2^{-\|z\|} = 1$ and $T_y(xz) \neq \perp$ for all $z \in Z$. We say (finite or infinite) string $\alpha \in \{0, 1\}^{*\infty}$ is to the “left” of $\beta \in \{0, 1\}^{*\infty}$, with $\alpha \triangleleft \beta$, if there exists $x \in \{0, 1\}^*$ such that $x0 \sqsubseteq \alpha$ and $x1 \sqsubseteq \beta$. A machine T is left-total if for all auxiliary strings $\alpha \in \{0, 1\}^{*\infty}$ and for all $x, y \in \{0, 1\}^*$ with $x \triangleleft y$, one has that $T_\alpha(y) \neq \perp$ implies that x is total with respect to T_α .

For the remaining part of appendix A, we can and will change the universal self delimiting machine U into a universal left-total machine U' by the following definition. The algorithm U' enumerates all strings $p \in \{0, 1\}^*$ in order of their convergence time of $U(p)$ and successively assigns them consecutive intervals $i_p \subset [0, 1]$ of width $2^{-\|p\|}$. Then U' outputs $U(p)$ on input p' if the open interval corresponding to p' and not that of $(p')^-$ is strictly contained in i_p . The open interval in $[0, 1]$ associated with p' is $((p')2^{-\|p'\|}, ((p')+1)2^{-\|p'\|})$ where (p) is the value of p in binary⁴. The same definition applies for the machines U'_α and U_α , over all $\alpha \in \{0, 1\}^{*\infty}$.

Without loss of generality, the complexity terms of section 3.3 are defined in this section with respect to the universal machine U' . (Not sure if this needs to be expanded upon). The infinite border sequence $\mathcal{B} \in \{0, 1\}^\infty$ represents the unique infinite sequence such that all its finite prefixes have total and non total extensions. The term “border” is used because for any string $x \in \{0, 1\}^*$, $x \triangleleft \mathcal{B}$ implies that x total with respect to U' and $\mathcal{B} \triangleleft x$ implies that U' will never halt when given

⁴For example, the value of both strings 011 and 0011 is 3. The value of 0100 is 4.

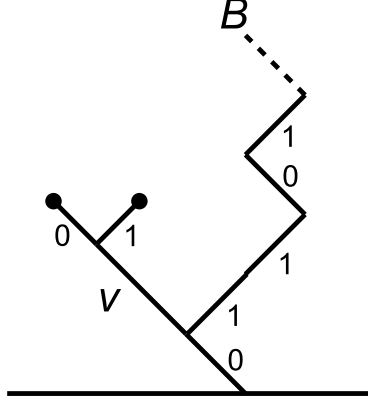


Figure 1: The above diagram represents the domain of the universal left-total algorithm U' , with the 0 bits branching to the left and the 1 bits branching to the right. The strings in the above diagram, $0v0$ and $0v1$, are halting inputs to U' with $U'(0v0) \neq \perp$ and $U'(0v1) \neq \perp$. So $0v$ is a total string. The infinite border sequence $\mathcal{B} \in \{0, 1\}^\infty$ represents the unique infinite sequence such that all its finite prefixes have total and non total extensions. It is the binary expansion of Chaitin's omega, $\Omega \in \mathbb{R}$.

x as an initial input. Figure 1 shows the domain of U' with respect to \mathcal{B} . The border sequence \mathcal{B} is the binary expansion of Chaitin's Ω , and thus random and computable from \mathcal{H} .

Proposition A.1. *For border prefix $b \sqsubseteq \mathcal{B}$, $\|b\| <^+ \mathbf{K}(b)$ and $\mathbf{K}(b|\mathcal{H}) <^+ \mathbf{K}(\|b\|)$.*

Proposition A.2. *If $b \in \{0, 1\}^*$ is total and b^- is not total, then b^- is a border prefix, with $b^- \sqsubseteq \mathcal{B}$.*

The following lemma shows that non-stochastic strings x are “exotic,” i.e. have high $\mathbf{I}(x; \mathcal{H})$ information with the halting sequence.

Lemma A.3. *For $\lambda(n) = O(n)$, $x \in \{0, 1\}^*$, $\chi_\lambda(x) <^{\log} \mathbf{I}(x; \mathcal{H})$.*

Proof. Let $U'(x^*) = x$, $\|x^*\| = \mathbf{K}(x)$, and v be the shortest total prefix of x^* . We define the primitive probability measure Q such that $Q(a) = \sum_w 2^{-\|w\|} [U'(vw) = a]$. Thus Q is computable relative to v . In addition, since $v \sqsubseteq x^*$, one has the lower bound $Q(x) \geq 2^{-\|x^*\| + \|v\|} = 2^{-\mathbf{K}(x) + \|v\|}$. Therefore

$$\begin{aligned}
\mathbf{d}(x|Q, v) &= \lfloor -\log Q(x) \rfloor - \mathbf{K}(x|v) \\
&\leq \mathbf{K}(x) - \|v\| - \mathbf{K}(x|v) \\
&<^+ (\mathbf{K}(v) + \mathbf{K}(x|v)) - \|v\| - \mathbf{K}(x|v) \\
&<^+ (\|v\| + \mathbf{K}(\|v\|) + \mathbf{K}(x|v)) - \|v\| - \mathbf{K}(x|v), \\
\mathbf{d}(x|Q, v) &<^+ \mathbf{K}(\|v\|).
\end{aligned} \tag{23}$$

Since v is total and v^- is not total, by proposition (A.2), v^- is a prefix of the border sequence

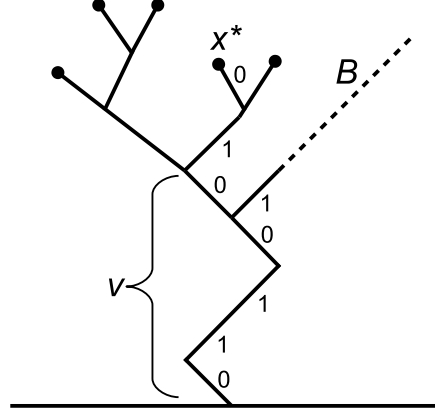


Figure 2: The above figure shows an example of the domain of left-total U' with the terms used in lemma A.3. $x^* = 0110010$ and $v = 01100$. Since v is total and v^- is not, v^- is a prefix of the border sequence \mathcal{B} . In the above example, assuming all halting extensions of v produce a unique output, $|\text{Support}(Q)| = 5$, and $Q(x) = 2^{-\|x^*\| + \|v\|} = 0.25$.

\mathcal{B} (see Figure 2). In addition, Q is computable from v . Therefore

$$\begin{aligned} \mathbf{K}(x|\mathcal{H}) &<^+ \mathbf{K}(x|Q) + \mathbf{K}(Q|\mathcal{H}) \\ &<^+ \mathbf{K}(x|Q) + \mathbf{K}(v|\mathcal{H}) \\ &<^+ -\log Q(x) + \mathbf{K}(\|v\|) \end{aligned} \tag{24}$$

$$\begin{aligned} &<^+ \mathbf{K}(x) - \|v\| + \mathbf{K}(\|v\|), \\ \|v\| &<^+ \mathbf{K}(x) - \mathbf{K}(x|\mathcal{H}) + \mathbf{K}(\|v\|), \\ \|v\| &<^{\log} \mathbf{I}(x; \mathcal{H}). \end{aligned} \tag{25}$$

Equation (24) is due to proposition (A.1). Since Q is computable from v , one gets $\chi_\lambda(x) <^+ \mathbf{K}(v) + \lambda(\max\{\mathbf{d}(x|Q, v), 1\}) <^+ \|v\| + \mathbf{K}(\|v\|) + \lambda(\max\{\mathbf{d}(x|Q, v), 1\})$. Due to equation 23 and the upper bound on λ , one gets $\chi_\lambda(x) \leq \|v\| + O(\mathbf{K}(\|v\|)) <^{\log} \|v\|$. Due to equation 25, one gets $\chi_\lambda(x) <^{\log} \mathbf{I}(x; \mathcal{H})$. \square

Our proofs separate the enumerative and combinatorial arguments using an approximation of \mathbf{m} . For all total strings $b \in \{0, 1\}^*$, we define the semimeasure $\mathbf{m}_b(x) = \sum \{2^{-\|p\|} : U'(p) = x, p \triangleleft b \text{ or } b \sqsubseteq p\}$. If b is not total then $\mathbf{m}_b(x) = \perp$ is undefined. Thus the algorithmic weight of a string x is approximated using programs that either extend b or are to the left of b .

Theorem 7.

For primitive map f , $\min_{a \in \text{Dom}(f)} f(a) + \mathbf{K}(a) <^{\log} -\log \sum_{a \in \text{Dom}(f)} \mathbf{m}(a)2^{-f(a)} + \mathbf{I}(\langle f \rangle; \mathcal{H})$.

Proof. Let $s = \lceil 1 - \log \sum_{a \in \text{Dom}(f)} \mathbf{m}(a)2^{-f(a)} \rceil$ and let $S(z) = \lceil -\log \sum_{a \in \text{Dom}(f)} \mathbf{m}_z(a)2^{-f(a)} \rceil$ be a partial recursive function from strings to rational numbers. S is defined solely on total strings, where $S(z) \neq \perp$ iff z is total. For total strings z, z^- , one has that $\mathbf{m}_{z^-}(x) \geq \mathbf{m}_z(x)$ and therefore $S(z^-) \leq S(z)$. Let b be the shortest total string with the property that $S(b) < s$. This implies $S(b^-) = \perp$ and thus b^- is not total. So by proposition (A.2), $b \sqsubseteq \mathcal{B}$ is a prefix of border. Theorem 6, conditioned

on b , with $m(a) = \mathbf{m}_b(a)$, provides $a \in \mathbb{W}$ such that $\mathbf{K}(a|m, b) + f(a) <^{\log} s + \chi(f|m, b)$. Since $\mathbf{K}(m|b) = O(1)$, we have equation (26). Lemma (A.3), conditional on b , results in equation (27), with

$$\mathbf{K}(a|b) + f(a) <^{\log} s + \chi(f|b), \quad (26)$$

$$\mathbf{K}(a|b) + f(a) <^{\log} s + \mathbf{I}(f; \mathcal{H}|b), \quad (27)$$

$$\mathbf{K}(a|b) + f(a) <^{\log} s + \mathbf{K}(f|b) - \mathbf{K}(f|b, \mathcal{H}). \quad (28)$$

Using the fact that $\mathbf{K}(a) <^+ \mathbf{K}(a|b) + \mathbf{K}(b)$, we get $\mathbf{K}(a) - \mathbf{K}(b) <^+ \mathbf{K}(a|b)$, and combined with equation (28), we get equation (29). Equation (30) is due to the chain rule $\mathbf{K}(b) + \mathbf{K}(f|b) <^{\log} \mathbf{K}(f) + \mathbf{K}(b|f)$. Equation (31) follows from the inequality $\mathbf{K}(f|\mathcal{H}) <^+ \mathbf{K}(f|b, \mathcal{H}) + \mathbf{K}(b|\mathcal{H})$.

$$\mathbf{K}(a) + f(a) <^{\log} s + \mathbf{K}(b) + \mathbf{K}(f|b) - \mathbf{K}(f|b, \mathcal{H}), \quad (29)$$

$$\mathbf{K}(a) + f(a) <^{\log} s + \mathbf{K}(f) + \mathbf{K}(b|f) - \mathbf{K}(f|b, \mathcal{H}), \quad (30)$$

$$\mathbf{K}(a) + f(a) <^{\log} s + \mathbf{K}(f) + \mathbf{K}(b|f) - \mathbf{K}(f|\mathcal{H}) + \mathbf{K}(b|\mathcal{H}), \quad (31)$$

$$\mathbf{K}(a) + f(a) <^{\log} s + \mathbf{I}(f; \mathcal{H}) + (\mathbf{K}(b|f) + \mathbf{K}(b|\mathcal{H})). \quad (32)$$

The remaining part of the proof shows that $\mathbf{K}(b|f) + \mathbf{K}(b|\mathcal{H}) = O(\log(s + \mathbf{K}(b)))$. This is sufficient to prove the theorem due to its logarithmic precision and by the right hand side of the inequality of equation (29) being larger than $s + \mathbf{K}(b)$ (up to a logarithmic factor). Since b is a prefix of border, due to proposition (A.1), one gets that $\mathbf{K}(b|\mathcal{H}) < O(\mathbf{K}(\|b\|)) < O(\log \|b\|) < O(\log \mathbf{K}(b))$. Thus combined with equation (32) and also equation (29), one gets

$$\mathbf{K}(a) + f(a) <^{\log} s + \mathbf{I}(f; \mathcal{H}) + \mathbf{K}(b|f). \quad (33)$$

We now prove $\mathbf{K}(b|f) <^+ \mathbf{K}(s, \|b\|)$. This follows from the existence of an algorithm, that when given f , s , and $\|b\|$, computes $S(b')$ for all $b' \in \{0, 1\}^{\|b\|}$ ordered by \triangleleft , and then outputs the first b' such that $S(b') < s$. This output is b otherwise there exists total $b' \triangleleft b$, with $\|b'\| = \|b\|$, and $S(b') < s$. This implies the existence of a total string $b'' \sqsubseteq b$, $b'' \sqsubseteq b'$, $\|b''\| < \|b\|$ such that $S(b'') < s$. This contradicts the definition of b being the shortest total string with $S(b) < s$. So $\mathbf{K}(b|f) <^+ \mathbf{K}(s, \|b\|)$ and thus one gets the final form of the theorem, as shown below. Equation (34) is again due to the right and side of equation (29).

$$\begin{aligned} \mathbf{K}(a) + f(a) &<^{\log} s + \mathbf{I}(f; \mathcal{H}) + \mathbf{K}(s, \|b\|), \\ \mathbf{K}(a) + f(a) &<^{\log} s + \mathbf{I}(f; \mathcal{H}), \\ \min_{a \in \text{Dom}(f)} \mathbf{K}(a) + f(a) &<^{\log} - \log \sum_{a \in \text{Dom}(f)} \mathbf{m}(a) 2^{-f(a)} + \mathbf{I}(f; \mathcal{H}). \end{aligned} \quad (34)$$

□

References

A. Berthiaume, W. van Dam, and S. Laplante. Quantum Kolmogorov Complexity. *eprint arXiv:quant-ph/0005018*, 2000.

- G. Chaitin. A Theory of Program Size Formally Identical to Information Theory. *Journal of the ACM*, 22(3):329–340, 1975.
- S. Epstein and M. Betke. An Information Theoretic Representation of Agent Dynamics as Set Intersections. In *Proceedings of the Fourth Conference on Artificial General Intelligence*, volume 6830 of *LNAI*, pages 72–81, 2011.
- Samuel Epstein. All sampling methods produce outliers. *CoRR*, abs/1304.3872, 2013. URL <http://arxiv.org/abs/1304.3872>.
- Samuel Epstein and Leonid A. Levin. Sets have simple members. *CoRR*, abs/1107.1458, 2011-2014. URL <http://arxiv.org/abs/1107.1458>.
- P. Gács. Quantum algorithmic entropy. *Journal of Physics A Mathematical General*, 34:6859–6880, 2001.
- P. Gács, J. Tromp, and P. Vitányi. Algorithmic Statistics. *IEEE Transactions on Information Theory*, 47(6):2443–2463, 2001.
- A. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1:1–7, 1965.
- Leonid A. Levin. Sets have simple members. (repost). *CoRR*, abs/1403.4539, 2014. URL <http://arxiv.org/abs/1403.4539>.
- M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.
- A. Shen. The concept of (alpha,beta)-stochasticity in the Kolmogorov sense, and its properties. *Soviet Mathematics Doklady*, 28(1):295–299, 1983.
- A. Shen. Discussion on Kolmogorov Complexity and Statistical Analysis. *The Computer Journal*, 42(4):340–342, 1999.
- A. Shen. Game Arguments in Computability Theory and Algorithmic Information Theory. In *Proceedings of 8th Conference on Computability in Europe*, volume 7318 of *LNCS*, pages 655–666, 2012.
- R. Solomonoff. A Formal Theory of Inductive Inference, Part I. *Information and Control*, 7:1–22, 1964.
- N. Vereshchagin. On Algorithmic Strong Sufficient Statistics. In *Proceedings of 9th Conference on Computability in Europe*, volume 7921 of *LNCS*, pages 424–433. 2013.
- N. Vereshchagin and P. Vitányi. Algorithmic Rate Distortion Theory, 2004a. <http://arxiv.org/abs/cs.IT/0411014>.
- N. Vereshchagin and P. Vitányi. Kolmogorov’s Structure Functions and Model Selection. *IEEE Transactions on Information Theory*, 50(12):3265 – 3290, 2004b.

- N. Vereshchagin and P. Vitányi. Rate Distortion and Denoising of Individual Data using Kolmogorov Complexity. *IEEE Transactions on Information Theory*, 56:3438–3454, 2010.
- P. Vitanyi. Three Approaches to the Quantitative Definition of Information in an Individual Pure Quantum State. *eprint arXiv:quant-ph/9907035*, 1999.
- V. V'Yugin. On Randomness Defect of a Finite Object Relative to Measures with Given Complexity Bounds. *SIAM Theory of Probability and Its Applications*, 32:558–563, 1987.
- A. Zvonkin and L. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Math. Surveys*, page 11, 1970.